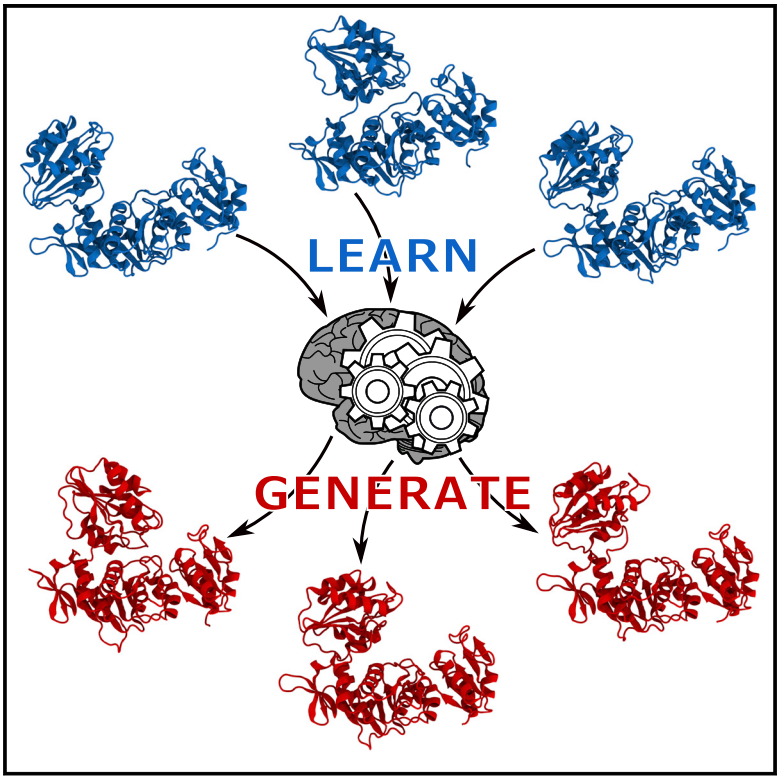
Short Article

Structure

Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space

Graphical Abstract



Authors

Matteo T. Degiacomi

Correspondence

matteo.t.degiacomi@durham.ac.uk

In Brief

Degiacomi presents a usage of generative neural networks for the characterization of the conformational space of proteins featuring domain-level dynamics. The network can generate protein-like structures and can be combined to a protein-protein docking algorithm to identify conformations close to the bound state.

Highlights

- A neural network is trained with protein atomic structures
- The trained network can generate new protein-like structures
- The neural network is used to generate candidate subunits for protein docking

Degiacomi, 2019, Structure *27*, 1034–1040
June 4, 2019
https://doi.org/10.1016/j.str.2019.03.018

CellPress

CellPress

Structure
Short Article

Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space

Matteo T. Degiacomi[1,2,*]
[1]Department of Chemistry, Durham University, South Road, Durham DH1 3LE, UK
[2]Lead Contact
*Correspondence: matteo.t.degiacomi@durham.ac.uk
https://doi.org/10.1016/j.str.2019.03.018

SUMMARY

Flexibility is often a key determinant of protein function. To elucidate the link between their molecular structure and role in an organism, computational techniques such as molecular dynamics can be leveraged to characterize their conformational space. Extensive sampling is, however, required to obtain reliable results, useful to rationalize experimental data or predict outcomes before experiments are carried out. We demonstrate that a generative neural network trained on protein structures produced by molecular simulation can be used to obtain new, plausible conformations complementing pre-existing ones. To demonstrate this, we show that a trained neural network can be exploited in a protein-protein docking scenario to account for broad hinge motions taking place upon binding. Overall, this work shows that neural networks can be used as an exploratory tool for the study of molecular conformational space.

INTRODUCTION

Function at the molecular level emerges from the arrangement of individual atoms and their associated dynamics. Specific interactions of simple molecules produce phenomena of increasing complexity, culminating with the finely tuned biological mechanisms that ultimately make life possible. Proteins are flexible molecules, and their dynamics are intimately connected to their function (Chu et al., 2013). The function can be modulated by conformational rearrangements in response to local environmental changes as diverse as changes in pH, temperature, or electrostatic potential, as well as binding to specific ligands such as ions, small molecules, lipids, or other proteins. As such, proteins should be seen not as a single static structure, but as a conformational ensemble featuring more or less accessible states.

To gain information about protein molecular functions, a broad range of techniques have been developed to interrogate their structure. X-ray crystallography provides precious structural evidence at atomic resolution, but its drawback is that it locks proteins in a single well-defined conformation within a crystal lattice. Atomic information can also be obtained via nuclear magnetic resonance. This technique also informs about dynamics and can sometimes identify multiple states, given that the molecule of interest is not too large. Other techniques report near-atomic or lower-resolution data, describing for instance the shape of the protein (e.g., electron microscopy, ion mobility-mass spectrometry, or small-angle X-ray scattering), or measuring specific interatomic distances (e.g., chemical crosslinking). Even considering this broad palette of techniques, studying the structure of a protein featuring multiple states is usually challenging, and even when a single conformation is present its thermal fluctuations may render data interpretation arduous.

Computational methods such as molecular dynamics (MD) simulations aim at characterizing molecular conformational space by iteratively generating new structures based on an initial, known atomic arrangement and a physical model of interatomic interactions. The structural ensemble produced by such sampling can be used to rationalize results of performed experiments, or help in obtaining information difficult to derive experimentally. Since simulations provide a discretized view of a continuous space, they will always be subject to the risk of missing key details owing to undersampling (Nemec and Hoffmann, 2017). This risk is usually low and acceptable when studying small or rigid proteins, and can be mitigated by running long simulations or exploiting enhanced sampling techniques. It becomes, however, increasingly severe the larger the molecular system under study and the slower its dynamics. A common, and extreme, scenario where MD is (in most cases) unsuitable is the prediction of the specific docking of multiple proteins into a complex. To tackle protein-protein docking, other sampling methods are typically exploited instead of relying on MD alone. These often utilize optimization engines exploring the roto-translational space of the binding partners, looking for the arrangements minimizing a specific scoring function (Kastritis and Bonvin, 2010; Zhang et al., 2016). It is acknowledged that accounting for molecular flexibility beyond the movement of amino acid side chains in docking processes is often key to producing suitable results (Zacharias, 2010) (Lensink et al., 2017). To account for conformational changes at backbone level, flexibility is typically accounted for by refining rigidly docked poses with methods such as energy minimization, MD or Monte Carlo (Schindler et al., 2015) (Dominguez et al., 2003), or docking pregenerated alternative conformations (Degiacomi and Dal Peraro, 2013; Marze et al., 2017). Overall, for any modeling scenario, whether MD simulations or docking, it is critical for conformational spaces to be extensively sampled.

In this work we examine the use of deep learning, and more specifically generative neural networks, to enrich the sampling

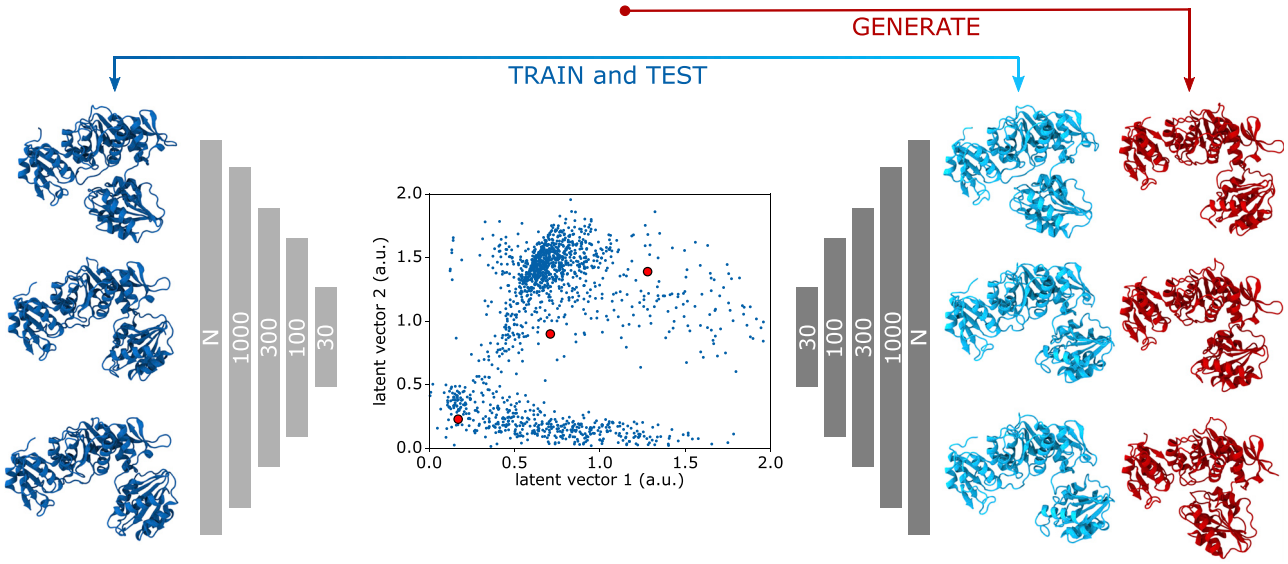


Figure 1. Autoencoder Structure

The autoencoder is a neural network composed of two parts: an encoder (light gray, with an input layer having a size equal to the degrees of freedom of the protein provided as input, followed by four hidden layers, with a decreasing amount of neurons noted in white) and a decoder (four dark-gray hidden layers, followed by a last layer having same size as the input). The first reduces protein atomic coordinates to a position in a low-dimensional space (blue points on the graph representing the so-called latent space) while the second converts such coordinates into a protein structure. The autoencoder is trained to encode-decode structures so that the difference between input (dark-blue proteins) and output (light-blue proteins) is minimized. After training, the decoder can be used to generate new protein structures (in red) from any coordinate within the latent space.

of molecular conformational space. While the usage of generative neural networks in image processing is widespread, their application to 3D point clouds is only a recent addition to the machine learning literature (Achlioptas et al., 2017). Proteins represent a particularly interesting application case in this area, since they do not feature a difficulty typical of raw 3D point clouds: as the position of their constituent atoms is constrained by covalent interactions, protein conformations can be interpreted as ordered sets of points. Generative neural networks have been recently proposed as a tool for the discovery of collective variables, useful to extract kinetic information from molecular simulations or to guide the sampling of poorly explored regions (Chen et al., 2018; Chiavazzo et al., 2017; Hernández et al., 2018; Mardt et al., 2018; Ribeiro et al., 2018).

Here we use the conformations generated by one or more protein MD simulations as examples to train an autoencoder (Hinton and Salakhutdinov, 2006) (Rumelhart et al., 1986). We demonstrate that autoencoders can generate new, realistic protein conformations complementing pre-existing data produced by MD simulations. We then show that a trained autoencoder coupled with a protein docking algorithm can be utilized to discover conformations closer to the bound state, given an ensemble of structures sampling the unbound state.

RESULTS

To generate low-dimensional representations of proteins' conformational space, we exploit an autoencoder (Figure 1). This is a type of neural network that attempts to first compress and then decompress a multidimensional input, so that the difference between input and output is minimized. The network is first trained with a collection of alternative molecular conformations. Its performance is then tested with a new set of conformations not previously used for training. The first part of the network, called the "encoder," passes the input signal (flattened Cartesian coordinates) thorough a series of fully connected hidden layers containing a decreasing number of neurons. This ultimately produces a low-dimensional representation of the input molecular structure, called the "latent vector." The values of the latent vector then become the input for a second series of hidden layers, this time with an increasing number of neurons, called the "decoder." The decoder expands the latent vector into an output that should be as similar as possible to the initial molecular structures passed through the encoder. In sum, the encoder allows casting a protein conformational space (possible atom positions in 3D space) into a non-linear, low-dimensional representation (latent space), whereas the decoder can convert the coordinates of such low-dimensional space into specific protein conformations. The system-specific way whereby data encoding and decoding takes place is determined by weights on connections between neurons, having values optimized during the training process.

Learning and Assessing Protein Conformational Space

While any coordinate within the latent space is associated with an atomic arrangement, not every coordinate will correspond to a plausible molecular structure. On one hand, while in general selecting values close to regions of the latent space sampled during the training of the network will usually yield a physically plausible model, we observed only a low correlation between the physical plausibility of a model and its distance in the latent space from points used as training examples. This is because

Table 1. Performance of Learning Algorithms on All Tested Protein Structures

Protein	PDB ID	d.o.f.[a]	Training Structures (n)[b]	MD RMSD (Å)[c]	Classifier Accuracy (%)[d]	Classifier Accept. (%)[e]	Scoring Function Accept. (%)[f]	Test Structures RMSD (Å)[g]	Test Structures Sec. Struct.[h]
Malate dehydrogenase*	1MLD	7,344	2,811	2.88	100	100	100	1.00 ± 0.13	95.9 ± 1.5
αB crystallin	2WJ7	1,857	2,829	4.79	99.6	99	100	1.12 ± 0.24	93.3 ± 1.9
Phospholipase A_2	1POA	1,286	4,690	3.84	98.5	98	94	0.89 ± 0.15	96.4 ± 1.7
Envelope glycoprotein*	1SVB	4,632	2,191	6.21	97.3	98	99	1.67 ± 0.35	94.0 ± 1.2
MurD, closed*	3UAG	5,124	2,507	3.80	99.6	100	100	1.14 ± 0.22	94.9 ± 1.5
MurD, open	1E0D	5,124	1,813	5.77	100	100	100	1.43 ± 0.50	93.9 ± 1.3
MurD, closed + open	3UAG 1E0D	5,124	4,320	10.22	100	100	100	0.90 ± 0.16	94.4 ± 2.0
HIV-1	1E6J	2,481	6,142	17.84	99.5	99	92	1.47 ± 0.50	96.0 ± 2.1

All cases were encoded in a 2D latent space. Cases indicated with an asterisk did not reproduce the training set's diversity.
See also Tables S1 and S2; Figures S2 and S3.
[a]Degrees of freedom, i.e., 3 times the amount of atoms.
[b]Quantity of structures used to train the autoencoder.
[c]Maximal RMSD within all structures in the simulation, reporting on the structural variability in the conformational space.
[d]Random Forests classifier accuracy after training.
[e]Percentage of reconstructed test structures accepted by the Random Forests classifier.
[f]Percentage of reconstructed test structures with a scoring function penalty equal to zero.
[g]RMSD of test structures against their reconstructed counterparts.
[h]Percentage of secondary structure elements in test structures matching those in their reconstructed counterparts.

distances in latent space do not usually linearly correlate with distances in the 3-dimensional (3D) Cartesian space (Figure S1). Generally it is difficult to determine what the dimensions in latent space represent in terms of motion in the 3D space. Given the difficulty in predicting whether a coordinate in the latent space will be associated with a physically plausible structure, we defined two methods to determine whether a model generated by the autoencoder should be considered as valid or not.

To quickly assess whether structures generated from the latent space should be considered plausible, we tested their ability to fool another learning algorithm, trained to determine whether an atomic arrangement provided as input is protein-like or not. To do so, we adopted a Random Forests (RF) classifier (Breiman, 2001) trained on two classes of data: structures extracted from an MD simulation ("correct") and structures from the same simulation, with a small amount of noise added to atomic coordinates (see Supplemental Information and Figure S2). Furthermore, we also defined two heuristic criteria, seen as necessary conditions, to assess whether models feature atoms excessively far from all the others or close to any other ("stretching" and "compression," respectively, see STAR Methods).

We first selected a small pool of proteins of different size and shape to assess whether the autoencoder would be able to reconstruct their conformational space, and the RF classifier to distinguish between good and bad ones. For this test, we selected malate dehydrogenase (a 33-kDa homodimer), αB crystallin (a rigid 18-kDa homodimer), phospholipase A_2 (13 kDa, featuring two long random coil regions 15 and 25 amino acids long, respectively) and encephalitis virus envelope glycoprotein (an elongated 43-kDa protein, approximately a cylinder three times longer than wide). For all proteins, we ran MD simulations >200 ns long, and trained the autoencoder to compress their conformational space (backbone and Cβ atoms) down to a 2-dimensional (2D) space. Since, unlike the general case of 3D point clouds, proteins are represented by ordered sets of points, root-mean-square deviation (RMSD) is a usable metric to assess the similarity between autoencoder-generated models and their MD-generated counterpart. In all cases, test structures could be reconstructed with an RMSD <1.5 Å from the original, a deviation smaller than that observed within the MD simulations (Table 1). In these reconstructions, distributions of bond distances and angles closely matched those observed in input structures, with no significant difference between rigid and more flexible cases (Table S1). Furthermore, in each case the secondary structure of >93% of amino acids matched that of their input structure counterpart (Table 1). We performed these same analyses on models generated using principal components analysis (PCA), whereby each simulation is projected on a 2D eigenspace and linear combinations of the two eigenvectors are used to regenerate a molecular structure. Concerning bonds and angles, we observed that PCA is more accurate than our autoencoder for rigid targets, whereas the opposite is true for flexible ones (Table S1). Concerning secondary structure, we found PCA to perform equally or marginally better than our autoencoder in all but the most flexible of test cases (Table S2).

To automatically verify whether the reconstructions of an autoencoder could be considered as new plausible protein conformations, we tested them using both our scoring function and the RF classifier. In all cases >98% of reconstructed structures were considered valid by the classifier, and all but one case had >97% structures featuring no stretching or compression (Table 1). This shows that the autoencoder can produce protein conformations close to the conformational space provided as input.

In three of our test sets, featuring the most large and rigid proteins (i.e., with structures in the training set having a low average RMSD), autoencoders learned to associate the same average

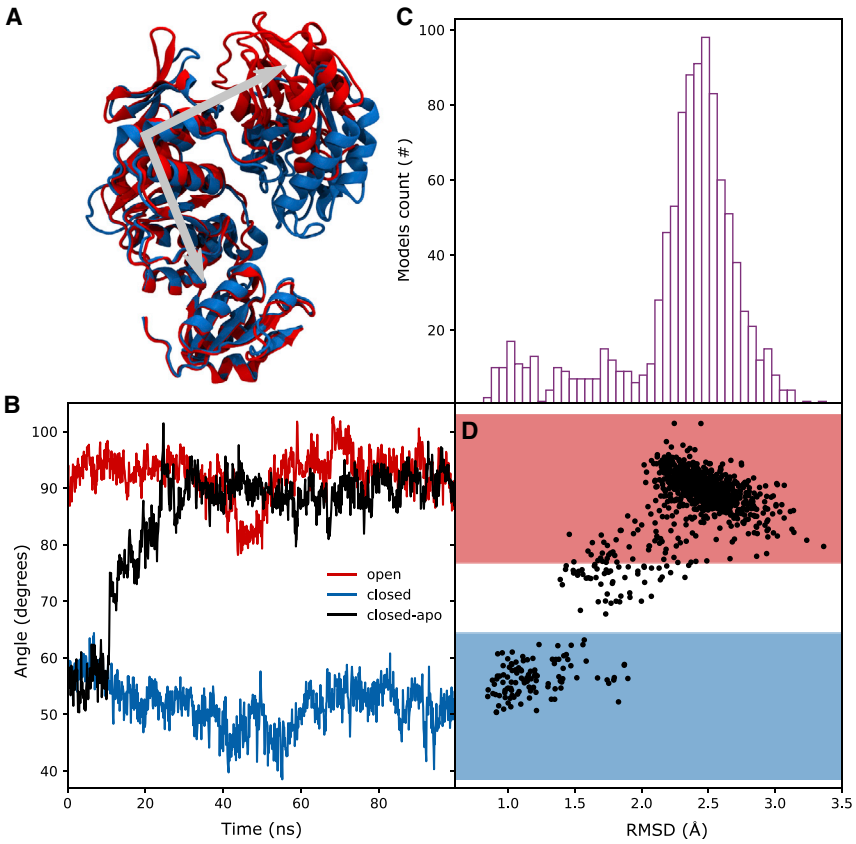


Figure 2. Reconstruction of MurD Closed-to-Open Transition

(A) For all MurD simulations, we calculated the angle between the distal ends of domain 2 and domain 3 (formed by the two gray vectors).

(B) The closed bound state (blue) and the open unbound one (red) preserve their specific interdomain orientations, whereas removing ligands from the closed state leads the protein to convert into the open state (black).

(C) The palatinate histogram shows the RMSD of encoded-decoded closed-apo structures with respect of the original structure.

(D) We trained the autoencoder with frames from open and closed states and asked it to encode-decode structures from the closed-apo simulation. For each reconstruction, we calculated the RMSD with respect to the input. We report the RMSD against the domains' angle. The closed state is more accurately reconstructed than the open state. Frames in white regions have an opening angle different from anything seen in both open (red zone) and closed (blue zone) simulation. They are nevertheless reconstructed with an RMSD around 2 Å. See also Figure S1.

protein structure with each input conformation (Table S1; Figures S2 and S3). Thus, we observe that effective autoencoders are easier to train with flexible proteins.

Representing a Multistate System

As a more difficult test case, we selected MurD, a 47-kDa ATP-driven ligase responsible for the biosynthesis of a bacterial peptidoglycan precursor (UDP-N-acetylmuramoyl-L-alanyl-D-glutamate). MurD has been crystallized in both open, unbound, state (PDB: 1E0D [Bertrand et al., 2000]) and closed state (PDB: 3UAG [Bertrand et al., 1999]), bound to GDP and UDP-N-acetylmuramoyl-L-alanyl-D-alanine. Comparing these structures shows that ligand binding causes one of the three globular domains of MurD (residues 300–439) to rearrange (Figure 2A). We performed three MD simulations: one of the open state, one of the closed state, and one of the closed state with its ligands removed (herein called closed-apo). In 200 ns the first two simulations maintained their specific domains' arrangement, while the third transitioned from the closed to the open state, sampling conformations being observed in neither the closed nor the open simulations (Figure 2B).

We trained the autoencoder with structures from only open and closed simulations, asking it to compress their 5,124 degrees of freedom down to a 2D latent vector. Subsequent testing with a set of 100 randomly selected structures not used for training revealed that the RMSD between input and reconstructed structures was 0.9 ± 0.2 Å, with a best case of 0.7 Å and a worst case of 1.7 Å. We then tested whether the autoencoder trained with only open and closed conformation would be able to correctly reconstruct 1,000 structures from the closed-apo conformation. Comparison of input and encoded-decoded structures revealed an average RMSD of 2.2 ± 0.5 Å, with the best case of 0.8 Å and the worst case of 3.4 Å. Importantly, we were also able to obtain structures at the transition between closed and open state, i.e., different from anything submitted as example for training. These were reconstructed with an RMSD of 1.9 ± 0.3 Å (Figures 2C and 2D). We also trained the autoencoder with only structures from the open state and then asking it to reconstruct closed-state structures. Reconstructed molecules were of poor quality (RMSD of closed structures >4.2 Å) indicating that, at least in its present form, the neural network is not capable of extrapolating new states but is instead suitable to interpolate between collections of known states and only extrapolate locally, beyond the front of structures sampled by the MD simulation (Figure S1).

Docking Proteins with Flexibility

As a final test case, we selected the HIV-1 hexameric capsomer, for which the crystal structure of both hexamer (PDB: 3MGE [Pornillos et al., 2010]) and monomer (PDB: 1E6J [Monaco-Malbet et al., 2000]) have been solved. The RMSD between the monomeric structure and each of the subunits part the assembled state is very large, 10.5 Å. This system has been already used as a docking test case in our previous work (Degiacomi and Dal Peraro, 2013). Therein we have shown that when the unbound state of HIV-1 capsomer is studied by MD, conformations featuring higher similarity to the bound states are found (3.2 Å RMSD). These conformations were submitted to a docking algorithm implemented using our POW^{ER} optimization environment. We demonstrated that POW^{ER} could automatically select conformations close to the bound state as most suitable candidates to

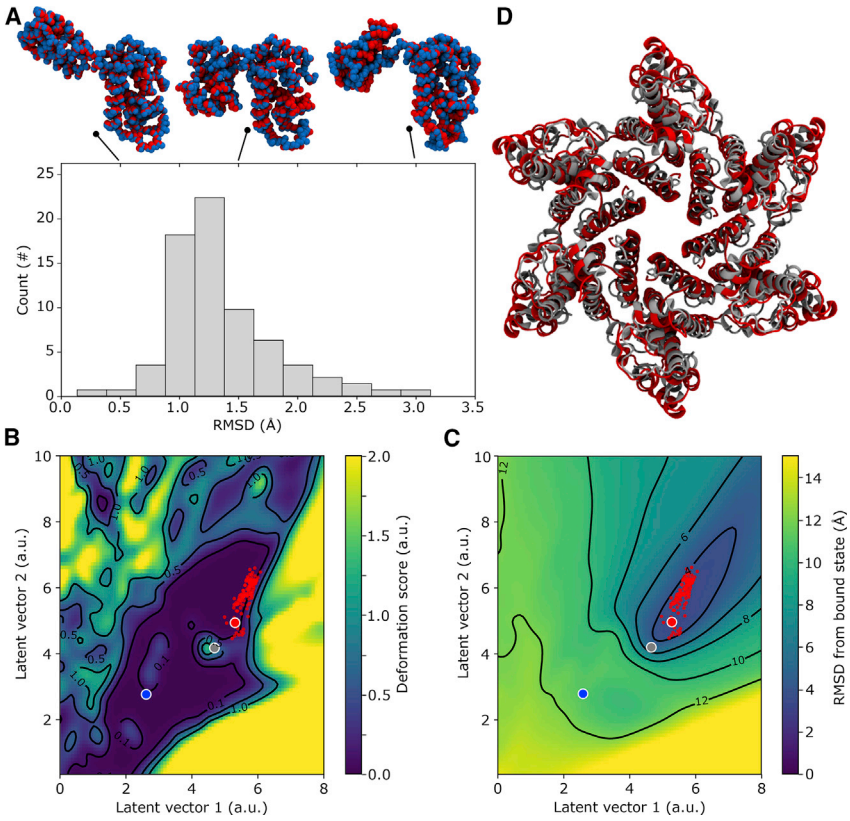


Figure 3. Protein-Protein Docking Using Structures Dynamically Generated by the Autoencoder

(A) Quality of 100 test set structures generated by the autoencoder trained with a simulation of HIV-1 capsomer's monomeric state. The histogram shows the distribution of RMSD between original and reconstructed structures. The upper panel shows three examples of reconstructions are shown (in blue the original structures, in red the predicted ones). Only atoms generated by the autoencoder, i.e., Cα, C, N, and Cβ, are shown. Examples show the best (0.5 Å RMSD), average (1.5 Å RMSD), and worst (3.0 RMSD Å) models produced.

(B) Two-dimensional latent space colored as a function of the model quality to which each position is associated, calculated as a sum of our compression and stretching heuristics. Large regions of the latent space correspond to plausible structures. Blue circle represents the encoded position of HIV-1 capsomer's monomeric crystal structure. Gray circle represents the position of a monomer extracted from hexameric HIV-1 capsomer structure (not used for training). Its associated reconstruction features a small amount of deformation. Small red circles represent the positions of structures selected by the docking algorithm as good candidates of HIV-1 capsomer's bound state. The red circle shows the position of the monomer used to build the model of HIV-1 capsomer with smallest RMSD from the known crystal structure.

(C) Two-dimensional latent space colored as a function of the RMSD against HIV-1 capsomer's bound state to which each position is associated. The RMSD between unbound and bound state is large, while a region close to the position of the bound state is associated to models with a small RMSD (<3 Å) from it.

(D) Superimposition of HIV-1 capsomer's hexameric crystal structure (in gray) and the best model generated by our docking algorithm (in red) leveraging on structures generated on demand by the trained autoencoder.

generate hexameric models, by augmenting its search space with the conformational space of molecular subunits described as coordinates into a low-dimensional eigenspace. Although promising, the limitation of such an approach is that the docking algorithm will provide results only as good as the best structure within the provided conformational ensemble. Furthermore, the optimization engines currently available within POW^{ER} are not profiled to handle discretized search space cases, meaning that we can expect their performance to be suboptimal.

We trained an autoencoder with a 2D latent vector on a microsecond-long simulation of the unbound state. One hundred randomly selected test structures were subsequently reconstructed with an RMSD of only 1.5 ± 0.5 Å (Figure 3A) from the original. We then characterized the whole latent space by generating all structures on a 200 × 200 sampling grid. Within the produced models we found that several were both acceptable (no penalty from both scoring function and RF classifier, Figure 3B), and with an RMSD from the bound state smaller than the best model available within the simulation (2.7 Å, against 3.2 Å from the MD simulation, Figure 3C). This shows that the latent space trained with unbound structures can describe protein conformations more suitable to form a complex than any of the structures with which it was trained.

We then assessed whether POW^{ER} would be able to identify such structures in the process of docking six HIV-1 monomers into a complex using two distance restraints. For this, we developed a new POW^{ER} module, leveraging on the trained autoencoder to generate candidate structures for docking. Thus, the search space to generate the HIV-1 hexamer according to a circular symmetry was 6-dimensional: two coordinates within the 2D latent space, three rotation angles to determine the orientation of each subunit within the complex, and one radius of the circular assembly (see STAR Methods). The docking protocol produced 151 models, from which 23 representatives were clustered. The best model in the top 10 had a Cα RMSD of 3.8 Å from the known complex, and the best model overall (rank 15) had an RMSD of 3.3 Å (Figure 3D). Interestingly, the subunits composing this model had an RMSD of 2.8 Å from the known bound state, lower than the best structure present in the MD simulation (3.2 Å). This result shows that, in cases where conformational selection plays a major role, coupling autoencoders and general optimization algorithms may help in predicting the structure of a protein assembly with subunits undergoing substantial concerted motions.

DISCUSSION

Herein, we have used autoencoders trained on structures from MD simulations as a tool for enriching the sampling of molecular conformational space. We have shown that their

low-dimensional latent space can be used to produce new molecular structures that are, from a geometric perspective, plausible. The dimensionality reduction capabilities of autoencoders are connected to the non-linear nature of their latent space. As in the case of other non-linear dimensionality reduction techniques (Das et al., 2006; Kim et al., 2015; Tribello et al., 2012), this leads to an enhanced capability of correctly capturing the complex movement of covalently bonded atoms as they explore the molecular conformational space. While our autoencoder appears to perform equally or slightly worse than PCA against rigid proteins, it performs better against more flexible ones. This indicates that the dimensionality reduction of an autoencoder is less affected than that of PCA by the complexity of protein dynamics (given a reduction to the same amount of degrees of freedom, here two). Since it is non-trivial to predict whether a coordinate in the latent space will be associated with a plausible protein structure, we adopted quick methods to assess whether a newly generated protein 3D structure can be considered as geometrically valid. Testing generated models with an RF classifier indicates that the autoencoder can produce protein structures considered correct by a learning algorithm trained to detect minimal discrepancies from ordinary atomic arrangements.

We were able to show that the latent space described by a trained autoencoder can be leveraged to extract information usable in contexts where extensive sampling is critical. In its current embodiment, the autoencoder is capable of interpolating between structures within the training set and to push the front beyond what has been observed in simulation, although this extrapolation capability is only local and does not include the discovery of new states. As an application example, we have shown how the autoencoder can be exploited to identify structures usable in a protein-protein docking scenario, implemented here in the POW^{ER} optimization engine. As the autoencoder does better at describing concerted motions (e.g., hinge motions) than at capturing subtle local fluctuations, it is most suitable to handle cases featuring domain-level rearrangements. In this context, we propose a simple model scoring function that is quick and effective when used by an optimization engine geared to minimize continuous fitness functions. The core idea behind all currently presented protein docking applications within POW^{ER} is that to rapidly create a first coarse subunit arrangement, an accurate energy function is not necessary if experimental data can be used as a guide. Suitable complexes generated in such a manner can be refined and reranked with more expensive and accurate computational techniques in a second step. Exploiting structures generated by neural networks for docking perfectly fits within this philosophy: as these feature plausible arrangements for backbone and Cβ atoms, missing side-chain atoms can be derived and refined a posteriori on the basis of standard force-field parameters, as demonstrated in our model secondary structure analysis. Further applications of our sampling approach lay in areas where experimental data report on ensemble quantities, such as chemical crosslinking.

In our tests, we have observed that it is easier to obtain an autoencoder capable of producing a range of different structures when the training set features a flexible protein. Although this shows that autoencoders work appropriately in cases where they can be most useful, this phenomenon remains undesirable. Preventing generative neural networks to become incapable of reproducing the training set's diversity is a subject of active research in the machine learning community (Arjovsky et al., 2017). Collecting further test cases will enable determining whether there exists a single neural network architecture and training protocol yielding the best performance, or whether vice versa custom solutions (i.e., testing multiple possible network structures and training protocols) are required, on a per case scenario, to obtain optimal performance. Given a sufficiently large dataset, it may be possible to train a general neural network for molecular modeling that could be quickly trained via transfer learning to tackle a specific conformational space sampling problem.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- CONTACT FOR REAGENT AND RESOURCE SHARING
- METHOD DETAILS
  - Molecular Dynamics Simulations
  - Analysis of MurD Opening Angle
  - Autoencoder Design
  - Random Forests Classifier
  - Protein Docking
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Random Forests Benchmarking
  - Autoencoder Benchmarking
- DATA AND SOFTWARE AVAILABILITY

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.str.2019.03.018.

ACKNOWLEDGMENTS

I wish to thank Dr. Chris Willcocks for useful discussions and Dr. Valentina Erastova for critically reviewing the manuscript. The work was supported by the Engineering and Physical Sciences Research Council (EP/P016499/1).

AUTHOR CONTRIBUTIONS

Conceptualization, Methodology, Investigation, Software, Writing – Original Draft, Writing – Review & Editing, Data Curation, Funding Acquisition: M.T.D.

DECLARATION OF INTERESTS

The author declares no competing interests.

Received: July 3, 2018
Revised: January 25, 2019
Accepted: March 25, 2019
Published: April 25, 2019

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow: a system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), pp. 265–284.

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. (2017). Learning representations and generative models for 3D point clouds. ArXiv, 1707.02392.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning. Proc. Mach. Learn. Res *70*, 214–223.

Bertrand, J.A., Auger, G., Martin, L., Fanchon, E., Blanot, D., Le Beller, D., Van Heijenoort, J., and Dideberg, O. (1999). Determination of the MurD mechanism through crystallographic analysis of enzyme complexes. J. Mol. Biol. *289*, 579–590.

Bertrand, J.A., Fanchon, E., Martin, L., Chantalat, L., Auger, G., Blanot, D., Van Heijenoort, J., and Dideberg, O. (2000). "Open" structures of MurD: domain movements and structural similarities with folylpolyglutamate synthetase. J. Mol. Biol. *301*, 1257–1266.

Breiman, L. (2001). Random forests. Mach. Learn. *45*, 5–32.

Chen, W., Tan, A.R., and Ferguson, A.L. (2018). Collective variable discovery and enhanced sampling using autoencoders: innovations in network architecture and error function design. J. Chem. Phys. *149*, 072312.

Chiavazzo, E., Covino, R., Coifman, R.R., Gear, C.W., Georgiou, A.S., Hummer, G., and Kevrekidis, I.G. (2017). Intrinsic map dynamics exploration for uncharted effective free-energy landscapes. Proc. Natl. Acad. Sci. U S A *114*, E5494–E5503.

Chollet, F. (2015). Keras, https://keras.io.

Chu, X., Gan, L., Wang, E., and Wang, J. (2013). Quantifying the topography of the intrinsic energy landscape of flexible biomolecular recognition. Proc. Natl. Acad. Sci. U S A *110*, E2342–E2351.

Das, P., Moll, M., Stamati, H., Kavraki, L.E., and Clementi, C. (2006). Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. Proc. Natl. Acad. Sci. U S A *103*, 9885–9890.

Degiacomi, M.T., and Dal Peraro, M. (2013). Macromolecular symmetric assembly prediction using swarm intelligence dynamic modeling. Structure *21*, 1097–1106.

Dominguez, C., Boelens, R., and Bonvin, A. (2003). HADDOCK: a protein–protein docking approach based on biochemical or biophysical information. J. Am. Chem. Soc. *125*, 1731–1737.

Eswar, N., Webb, B., Marti-Renom, M.A., Madhusudhan, M.S., Eramian, D., Shen, M.Y., Pieper, U., and Sali, A. (2007). Comparative protein structure modeling using modeller. Curr. Protoc. Protein Sci. *2*, 15–32.

Ge, R., Huang, F., Jin, C., and Yuan, Y. (2015). Escaping from saddle points—online stochastic gradient for tensor decomposition. In Conference on Learning Theory, pp. 797–842.

Hernández, C.X., Wayment-Steele, H.K., Sultan, M.M., Husic, B.E., and Pande, V.S. (2018). Variational encoding of complex dynamics. Phys. Rev. E *97*, 062412.

Hinton, G.E., and Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. Science *313*, 504–507.

Kabsch, W., and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers *22*, 2577–2637.

Kastritis, P.L., and Bonvin, A.M.J.J. (2010). Are scoring functions in protein-protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark. J. Proteome Res. *9*, 2216–2225.

Kim, S.B., Dsilva, C.J., Kevrekidis, I.G., and Debenedetti, P.G. (2015). Systematic characterization of protein folding pathways using diffusion maps: application to Trp-cage miniprotein. J. Chem. Phys. *142*, 085101.

Lensink, M.F., Velankar, S., and Wodak, S.J. (2017). Modeling protein-protein and protein-peptide complexes: CAPRI 6th edition. Proteins *85*, 359–377.

Maier, J.A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K.E., and Simmerling, C. (2015). ff14SB: improving the accuracy of protein side chain and backbone parameters from ff99SB. J. Chem. Theory Comput. *11*, 3696–3713.

Mardt, A., Pasquali, L., Wu, H., and Noé, F. (2018). VAMPnets for deep learning of molecular kinetics. Nat. Commun. *9*, 5.

Marze, N.A., Burman, S.S.R., Sheffler, W., and Gray, J.J. (2018). Efficient flexible backbone protein-protein docking for challenging targets. Bioinformatics *34*, 3461–3469.

Monaco-Malbet, S., Berthet-Colominas, C., Novelli, A., Battaï, N., Piga, N., Cheynet, V., Mallet, F., and Cusack, S. (2000). Mutual conformational adaptations in antigen and antibody upon complex formation between an Fab and HIV-1 capsid protein p24. Structure *8*, 1069–1077.

Nemec, M., and Hoffmann, D. (2017). Quantitative assessment of molecular dynamics sampling for flexible systems. J. Chem. Theory Comput. *13*, 400–414.

Phillips, J.C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R.D., Kalé, L., and Schulten, K. (2005). Scalable molecular dynamics with NAMD. J. Comput. Chem. *26*, 1781–1802.

Pornillos, O., Ganser-Pornillos, B.K., Banumathi, S., Hua, Y., and Yeager, M. (2010). Disulfide bond stabilization of the hexameric capsomer of human immunodeficiency virus. J. Mol. Biol. *401*, 985–995.

Ribeiro, J.M.L., Bravo, P., Wang, Y., and Tiwary, P. (2018). Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). J. Chem. Phys. *149*, 072301.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. Nature *323*, 533–536.

Schindler, C.E.M., de Vries, S.J., and Zacharias, M. (2015). iATTRACT: simultaneous global and local interface optimization for protein-protein docking refinement. Proteins *83*, 248–258.

Tribello, G.A., Ceriotti, M., and Parrinello, M. (2012). Using sketch-map coordinates to analyze and bias molecular dynamics simulations. Proc. Natl. Acad. Sci. U S A *109*, 5196–5201.

Wang, J., Wang, W., Kollman, P.A., and Case, D.A. (2006). Automatic atom type and bond type perception in molecular mechanical calculations. J. Mol. Graph. Model. *25*, 247–260.

Zacharias, M. (2010). Accounting for conformational changes during protein-protein docking. Curr. Opin. Struct. Biol. *20*, 180–186.

Zhang, Q., Feng, T., Xu, L., Sun, H., Pan, P., Li, Y., Li, D., and Hou, T. (2016). Recent advances in protein-protein docking. Curr. Drug Targets *17*, 1586–1594.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited Data		
Raw and analysed data	This paper	https://doi.org/10.15128/r26w924b81m
Software and Algorithms		
POW^{ER}	Degiacomi and Dal Peraro, 2013	http://lbm.epfl.ch/resources
Autoencoder and data analysis tools	This paper	https://doi.org/10.15128/r26w924b81m

CONTACT FOR REAGENT AND RESOURCE SHARING

Further information and requests for data will be fulfilled by the author, Matteo T. Degiacomi (matteo.t.degiacomi@durham.ac.uk).

METHOD DETAILS

Molecular Dynamics Simulations

Simulations were run with using the Amber ff14SB (Maier et al., 2015) on the NAMD (Phillips et al., 2005) molecular dynamics engine. For MurD closed state, ligands were parameterized using ANTECHAMBER (Wang et al., 2006). All atom types were found within existing ffSB14 parameters. Furthermore, the crystal structure of both MurD closed and open state featured two gaps (residues 221 to 224 and 242 to 244 for the closed, and 221, 222 and 183 to 188 in the open). We added all those missing regions using MODELLER (Eswar et al., 2007). The MurD closed-apo state was produced by removing the ligands from the closed state.

All proteins were solvated with TIP3P water and the resulting boxes neutralized with addition of Na^+ and Cl^- ions. The resulting systems energy minimized with 2000 conjugate gradient steps. We then performed 0.5 ns simulation with 2 fs time step (restraining every covalent bond with SHAKE) in the NPT ensemble, with all protein's C_α constrained by a harmonic potential. In all simulations Langevin dynamics were used to impose a temperature of 300 K, using a damping of 1 ps^{-1}. A constant pressure of 1 Atm was imposed via a Langevin piston having a period of 200 fs, and a decay of 50 fs. The systems were then further equilibrated in the NVT ensemble for 1 ns, after which production runs of variable length (between 0.1 and 1 μs) in the NPT ensemble were performed. In all simulation steps, particle mesh Ewald was used to treat long range electrostatic interactions, a cutoff distance of 12 Å was set on van der Waals interactions.

Analysis of MurD Opening Angle

The opening angle of MurD was calculated, in all simulations, between the centre of mass of three selections: residues 120-230 and 230-299 (distal ends of domain 2) and 299-437 (domain 3). The vectors connecting these three centres of mass are shown graphically with two gray arrows in Figure 2.

Autoencoder Design

The autoencoder was developed in Python 3.5 using the Keras package (Chollet, 2015), with Tensorflow backend (Abadi et al., 2016). In order to identify a suitable autoencoder structure and training protocol, we performed a systematic test using training and test sets generated from our MD simulations. From each simulation, one frame every 100 ps was extracted, selecting only Cα, C, N, and Cβ atoms to represent the proteins' backbone and sidechains directions. In the case of MurD, we combined 1913 structures from the open state and 2607 for the closed one. The coordinates of each dataset were first preprocessed: each simulation was aligned (by minimizing the RMSD from the first structure) and shifted so that atoms would only have positive coordinates. Finally, coordinates were normalizing between zero and one. The first and last layer of the autoencoder were *N*-dimensional, i.e. one dimension per protein degree of freedom. We tested two different networks, one featuring 3 encoding and 3 decoding layers (hereon "3-layer autoencoder"), and one featuring 5 encoding and 5 decoding layers (hereon "5-layer autoencoder"), all with 20% dropout. In both cases, we used a RELU activation function for each layer but the last one that was set as sigmoid, and used a binary cross-entropy loss function.

Random Forests Classifier

Determining whether an arrangement of atoms corresponds to a plausible protein conformation or not can be interpreted as a classification problem. A protein with N atoms can be represented as a single point in a 3xN dimensional space. The classifier divides this space into two regions, corresponding to plausible and not plausible protein conformations, respectively. To tackle this problem, we

adopted a Random Forests (RF) classifier. In short, RF is composed of an ensemble of decision trees (“weak learners”), trained based on examples of known class. After training, RF classifies an input structure based on the most voted within all its decision trees. Preliminary tests indicated that using more than 50 estimators did not improve the classifier’s performance. We therefore set 50 as number of learners, while maximal tree depth was left unbound. For every test case described in Main Text, we generated examples of unsuitable protein conformations by altering the coordinates of each MD-generated protein conformation with random noise (i.e. applying small displacements to each atom them, see Figure S2). 95% of MD-generated structures and 95% of altered structures were united and used as training set (structures selected at regular intervals), the remainder as test set. For each simulation, as for the autoencoder, only Cα, C, N, and Cβ were considered.

Protein Docking

To dock six HIV-1 subunits into a complex with POW^{ER}, we adopted a docking protocol previously described (Degiacomi and Dal Peraro, 2013). In summary, in order to build a circular hexamer, the search space was defined as three rotation angles defining the orientation of each monomer, the radius of the circle, and two dimensions defining coordinates in the latent space described by the autoencoder. To assemble a specific model, POW^{ER} would first require the autoencoder to generate the structure associated to a specific coordinate in the latent space, and then assemble the model according to desired rotations and radius values. Docking accounted only for atoms generated by the autoencoder, i.e. Cα, C, N and Cβ. The fitness function to be minimized featured a sum of terms including a 9-6 Lennard Jones potential to avoid steric clashes of backbone atoms, an error function assessing the matching with experimental data, as well as two terms, $S_{stretch}$ and $S_{compress}$, assessing the quality of the subunit generated by the autoencoder (“stretching” and “compression”, as defined in main text). These two latter components were designed to be continuous, in order to drive the optimization algorithm towards regions of the conformational space yielding plausible models. Let **d** the N dimensional vector reporting, for each atom, the distance to the closest neighbor.

$$S_{\text{stretch}}(\boldsymbol{d}) = \begin{cases} 0 & \text{if } \max(d_s) < 2 \\ \max(\boldsymbol{d}) - 2 & \text{otherwise} \end{cases} \quad \text{(Equation 1)}$$

$$S_{\text{compress}}(\boldsymbol{d}) = \begin{cases} 0 & \text{if } \min(\boldsymbol{d}) > 0.1 \\ 5 - \min(\boldsymbol{d}) * 50 & \text{otherwise} \end{cases} \quad \text{(Equation 2)}$$

Data used to guide the docking process was based on two loose distance restraints, i.e. a disulfide bridge (between Cys42 and Cys54) and a salt bridge (between Glu212 and Lys140) between neighbouring dimers. These had to be both plausible, i.e. have their respective amino acids at <5 Å. All models generated by POW^{ER} with fitness function smaller than zero (no clash, matching experimental data, plausible subunits) were accepted and clustered *via* a UPGMA hierarchical clustering, using a 2 Å cutoff. All reported RMSDs were calculated using the coordinates of all backbone and C_β atoms. The model shown in Figure 3D had the positions of all missing atoms reconstructed by Amber’s *tleap* tool.

QUANTIFICATION AND STATISTICAL ANALYSIS

Random Forests Benchmarking

To assess how accurately the RF classifier could discriminate between perturbed and unperturbed structures, we performed multiple training runs with variable levels of random noise (see Figure S2). We found that in all cases the classifier can discriminate with >99% accuracy a true structure from an altered one when noise level as low as 0.5 Å are applied (i.e., every degree of freedom altered with a random number uniformly distributed between 0 and 0.5 Å).

Autoencoder Benchmarking

For protein modelling test cases, we assessed the performance of both the 3-layer and 5-layer autoencoder using different sizes of the latent vector (2, 3, 4 neurons) and optimizer (Adam, SGD). We also tested the effect of training using different batch sizes (50, 100, 200, 300 examples per batch), whereby at every epoch the training set is randomly subdivided, and the neural network’s weights updated based on the combined performance against each item of a batch. This approach has been shown to be useful to escape local minima and improves the network’s generalization performance (Ge et al., 2015). Autoencoders performance was assessed in terms of RMSD of reconstructed test structures against original ones. For each combination of these parameter and network, we trained the resulting autoencoder three times for 500 epochs, for a total of 144 independent training runs per case (see Data S1). At every run, 100 randomly selected structures were separated and used for testing. We found that the Adam optimizer outperforms SGD, that 2 encoding neurons are in most cases enough to yield good performance, that 5 layers-deep networks perform marginally better, and that batch size does not have a significant effect on the overall performance. In main text we report the performance of the best 5-layers encoder and decoder trained with Adam, with a 2-dimensional latent vector, i.e. the highest data compression level we tested. For each run, we also calculated the pairwise RMSD of all structures used as test set, and that of the resulting reconstructed protein structure, to determine whether the autoencoder can reproduce the diversity in examples provided as input (see Figure S3).

Finally, we compared the secondary structure of encoded-decoded structures with their input counterparts. To do so, we assigned all missing atoms using Amber's *tleap* tool, and calculated the resulting model's secondary structure using DSSP (Kabsch and Sander, 1983). DSSP assigns secondary structure of each amino acid to one of eight different categories (seven kinds of structure, plus random coil).

DATA AND SOFTWARE AVAILABILITY

Data and software presented in this work is freely available for download in Durham University repository DRO-DATA.